NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CLANDESTINE MESSAGE PASSING IN VIRTUAL ENVIRONMENTS

by

Ryan Rippeon

September 2008

Co-Advisors: Gurminder Singh
Joseph Sullivan

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2008	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Clandestine Message Passing in Virtual Environments		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Ryan A. Rippeon			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

Virtual Environments (VEs) present a new challenge for government officials attempting to monitor computer networks for terrorist communication. VEs bring new dimensions to online communication through visual appearance and state maintaining servers. In this thesis, various VEs will be explored to study what current abilities and usage patterns exist. Once characteristics of the VEs are established, clandestine methods for passing information will be developed along with proof of concepts. Visual cues, steganography and autonomous bots will be examined. Monitoring techniques are then discussed to attempt observation and analysis of this information at various levels. The expectation is that these results will improve awareness and solidify an understanding of the more surreptitious capabilities present in these networked environments.

14. SUBJECT TERMS Message Passing, Virtual Environments, Steganography, Second Life, Internet Terrorism, Honeyworld, Sun MPK20, Clandestine Messages, Virtual Worlds, Massive Multiplayer Online	**15. NUMBER OF PAGES** 82
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

CLANDESTINE MESSAGE PASSING IN VIRTUAL ENVIRONMENTS

Ryan A. Rippeon
Lieutenant, United States Navy
B.S., United States Naval Academy, 2004

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN COMPUTER SCIENCE
and
MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS AND SIMULATION (MOVES)

from the

NAVAL POSTGRADUATE SCHOOL
September 2008

Author: Ryan Alexander Rippeon

Approved by: Gurminder Singh
Co-Advisor

CDR Joseph Sullivan
Co-Advisor

Mathias Kolsch
Chairman, MOVES Academic Committee

Peter Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Virtual Environments (VEs) present a new challenge for government officials attempting to monitor computer networks for terrorist communication. VEs bring new dimensions to online communication through visual appearance and state maintaining servers. In this thesis, various VEs will be explored to study what current abilities and usage patterns exist. Once characteristics of the VEs are established, clandestine methods for passing information will be developed along with proof of concepts. Visual cues, steganography and autonomous bots will be examined. Monitoring techniques are then discussed to attempt observation and analysis of this information at various levels. The expectation is that these results will improve awareness and solidify an understanding of the more surreptitious capabilities present in these networked environments.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

API – Application Programming Interface
BVH - Biovision hierarchical data file
Bot - Robot
COTS - Commercial Off The Shelf
CPU - Central Processing Unit
DoD – Department of Defense
DS – Dual Screen/Developers System
FPS – First Person Shooter
FAQ – Frequently Asked Questions
MMO – Massive Multiplayer Online
MOVES - Modeling Virtual Environments and Simulation
MTV – Music Television
NPS - Naval Postgraduate School
PAN – Personal Area Network
PSP - PlayStation Portable
RPG – Role-playing Game
SL - Second Life
SVN - Subversion
VE – Virtual Environments
vMTV – Virtual Music Television
WoW - World of Warcraft

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank everyone who helped me through this process. Gurminder Singh, a humble genius, has taught me some truly great classes and pushed me to think about the future in an innovative way. Joe Sullivan has been a supportive force throughout my time at NPS and has been a leader I admire. Chuck Villamarin and Tom Frey were always enthusiastic and overwhelmingly supportive of my efforts.

I appreciate the emphasis on education and perseverance instilled in me from my parents, Austin and Victoria. Even though my father was not here to see me reach this point, I still felt like he was watching and supporting. Carole and Dave have been there for me every time I needed them and helped me with big decisions. They are wonderful landlords. Kimberly, you always seemed to know what I was thinking; thank you for being a true friend and confidant - my best memories of Monterey will always include you. Anna, thanks for the laughs and support. Pat, I appreciated your help.

To all my other friends, family, and faculty who have helped, thank you. I could not have made it to this milestone without your support. NPS has been a great place and I will always remember the wonderful people I have been privileged to work with here.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MESSAGE PASSING

Passing messages from one individual to another has been a key aspect of humanity from the beginning of time. More often than not, those who master it succeed, while those who do not, fail.

Herodotus' The Histories describes a method used in the ancient world of passing a secret message. Histiaeus of Susa sends a message to a distant ally, Aristagoras of Miletus, urging him to revolt against the king. The roads were guarded by the king's forces, all messengers were being interrogated. Histiaeus shaved the head of his most trusted slave, tattooed the message on his head, and waited for the hair to grow back. Sent to the town of Miletus, the slave was given only one task: to tell Aristagoras to shave and examine his head. The plan worked, and the message was successfully passed on.

This method had a number of advantages at that time:

- Possessing no other knowledge, there was no information that the slave knew that could be of use to the interrogators.
- Shaving the head of a slave to tattoo a message, allowing the hair to grow back, then sending them on their way was a completely original idea. It strayed from the traditional written or memorized message that would be passed.
- Time to covertly send a message using this method was long, but, it matched the military campaign cycle of that age. Campaigns were planned on timelines with units of weeks and months, rather than the minutes and hours of today.

- Commanders then didn’t have the forward presence that we enjoy in modern times. The ability to download a video feed in encrypted real time from an Unmanned Aircraft System, half way around the world would seem like magic to people in the age that Herodotus was writing.

Message passing has certainly changed from these ancient world methods. Many innovations like the telegraph, radio, telephone, and television have come to pass since this time. Each has revolutionized the way people do business – both legal and illegal.

B. PROBLEM STATEMENT

Virtual Environments (VEs) are one of the newest popular tools for exchanging information on the Internet. They incorporate classic technologies like instant messenger, chat rooms, voice communication, file transfer, and social networks into a 3-dimensional virtual world. With a push toward different business models using distributed technology and corporate America shying away from the classic email task list, companies are increasingly exploring the promise of an advanced collaboration environment. Hardware has improved to the point where standard graphics cards and processors can handle three dimensional VEs. Bandwidth has also increased for many people – both at home and in the work environment – for effective real-time communication. All of these factors considered together paint the picture of a robust future for VE-based software.

With a favorable outlook for VEs in the future, the methods by which VE users pass messages need to be addressed. The potential exists for a terrorist or

subversive force to use it for coordination and planning of their attacks, while keeping law enforcement agencies in the dark.

C. SCOPE

The overall focus of this thesis is a threat analysis of the methods of passing messages in VEs.

This thesis will start by exploring traditional methods of message passing using computers. These will be adapted for VEs. New techniques that VEs bring to the table will be discussed. Interception of these new techniques will be evaluated. Finally, new ideas about monitoring will be examined in the VE setting.

D. ORGANIZATION OF THESIS

The thesis is organized in the following chapters.

Chapter I – Introduction. This chapter provides the motivation for the research, the questions asked, the level that the research will examine, and the method taken to answer the problems.

Chapter II – Background. This chapter covers a literature review of different aspects of message passing. Techniques, practices, influences and protocols will be covered, contributing to the understanding of messages and how they are intercepted on the Internet.

Chapter III – Innovation and Experimentation in VE Messaging. This chapter details the various options possible for transferring messages in VEs. Experiments will be conducted using different techniques of message passing.

Chapter IV – Analysis and Monitoring of Virtual Environments. With all the innovative techniques to pass messages that are available, ideas for monitoring will be described in this chapter.

Chapter V – Summary and Conclusions. This chapter presents lessons learned from the various experiments. It also offers suggestions for future experiments and studies that would help address issues in this area. This section also lists different research topics for the future.

Appendix A – Ruby Steganography Detection Source Code. This section gives the source code for a Ruby Script that will search a directory of files for steganographic messages, as described in Chapter III.

Appendix B – BVH Animation Code. This section gives the Biovision hierarchical data (BVH) format source code for an animation script compatible with Second Life, as described in Chapter III.

Appendix C – C# Bot Source Code. This section gives the source code for a C# Bot that is capable of direct interaction with Second Life. It was written using the libsecondlife API.

II. BACKGROUND

A. VIRTUAL ENVIRONMENTS

Virtual Environments (VEs) are emerging as the new game servers, business tools, and social networking applications of the future. Game companies have embraced VEs and use them on console systems like Xbox 360 and PlayStation 3 as managerial software to set up person to person matches for any one of the hundreds of titles. The business world has tested the waters with various early environments in an attempt to find better collaboration tools. VEs are being seen as a part of the Web 3.0 movement – a gradual confluence of technologies that will fundamentally change the way people use the Internet (Berman, 2007). Graphics cards that can support these 3D worlds are becoming more common and are even required for the newest version of the *Windows* Operating System (Microsoft, 2008). VEs are still in their infancy, but they show a great deal of promise if they properly develop alongside hardware and technological innovations.

The strength of virtual environments seems to rest in their ability to bring many of the message passing tools together in one enhanced application. Users can participate in meetings where their virtual persons – avatars - are seated around tables in virtual conference rooms. Communication uses a chat room based system or a more advanced voice over IP conference call. Avatars can send private instant messages back and forth. Tools can

even be used as if they were in the real world – virtual white boards can be written on and virtual objects manipulated for demonstrations.

The typical VE software model is a client and server-based architecture where client users log in to a VE running on a server. Many different variants exist for the VE software under this base client/server architecture.

B. VIRTUAL ENVIRONMENTS HISTORY

Maze War was a game created in 1974 and is considered by some to be the first digital virtual world. All the elements of modern gaming VEs were present – multi-user, different points of view, 3D graphics, bots, chat, and IM (Terdiman, 2006). Beginning in the early 1990s, attempts were made at creating large-scale, networked virtual worlds capable of running on high end personal computers or servers. The Naval Postgraduate School experimented with NPSNet, a low-cost real-time interactive simulation using the Distributed Interactive Simulation (DIS) Protocol (Hearne 1993). The Institute of Systems Science (ISS) computer think-tank at the National University of Singapore was also an early pioneer in the VE realm with their *History City* VE (Das, 1997).

Vernacular developments have occurred to compliment the VE technology evolution. Many terms like “avatar” and “metaverse” have been made popular in literature, most notably in Neal Stephenson’s 1992 novel *Snow Crash*. These terms have worked their way into every day society and are now common knowledge.

Due to the constraints of hardware and the networked nature of VEs, the commercial interest in VE technology was limited. Popularity increased slowly, and in different areas. Virtual environments depend heavily on the confluence of technologies for advancement – a stunning graphical application for a distributed VE is useless if the connection cannot support it.

Today VEs are another lucrative way to advertise and sell products. Larger corporations are creating environments that appeal to any and every interest group. MTV has an entire set of VEs to cater to many of their most popular television series. *The Hills*, *Real World*, *Pimp My Ride*, and others all have their own VEs where viewers can interact (MTV, 2008). Disney's *Pirates of the Caribbean* has an online VE based on the movies (Disney, 2008). The Army developed a first person shooter VE called *America's Army* (America's Army, 2008). These games can focus on a distinct group of people for their marketing strategies - children in the Disney VEs, teen girls in *The Hills*, high school students for *America's Army*.

C. BUSINESS OF VIRTUAL ENVIRONMENTS FOR CONSUMERS

Many different business models for VEs have emerged recently. Second Life has a free, open source client that logs into a server running closed source, unreleased software. Initial accounts are free; each additional account costs money. Virtual land can be purchased for a monthly fee. Recently, IBM reverse engineered their VE from the client side and built their own compatible server.

While not as robust as the Second Life server, an avatar has been transferred from the Second Life server to the IBM server (Reuters, 2008).

The World of Warcraft environment implements a VE that charges users a monthly subscription fee. Part of this supports a dedicated team of programmers in developing new content, technical support, and policing the environment.

Sun's MPK20 is a completely free, open source client and server set of software. Users can run both the client and server software to set up and manage their own VE on private equipment.

There are several other business models including hosting public conferences in VEs, selling VE server space for use in enterprises (Linden Labs 2008), and for e-commerce (Tode, 2007). The danger in selling virtual space for use by a consumer lies in the privacy that exists in such a system. The only oversight that currently exists rests with the company selling virtual areas. Even this level of oversight can be removed if a user sets up their own server-based VE.

D. VIRTUAL ECONOMIES

The economies of virtual environments have ties to the real world's economies. Second Life has its own currency, Linden Dollars, which have an exchange rate of approximately 250 Linden Dollars to every one U.S. Dollar (Linden Labs, 2008). This virtual money can be bought and sold on the Linden Exchange, LindeX, financial system. Economic meltdown occurred when false interest rates caused a run on the banking system's Linden dollars (Miller,

2008). Virtual economies that have no initial ties to real world economies eventually generate links that cross the bounds. World of Warcraft has an in-game monetary system where pieces of virtual gold can be earned by fighting monsters and completing quests. Gold can then be used to purchase upgraded virtual armor and improved items. By placing an in-game value on these items, an exchange market was established for their illegal sale (Hoglund 2008). A search on the internet will yield thousands of sites offering virtual gold and other items in exchange for real world currency. This is taken further when citizens of developing countries are paid a small base salary to earn virtual items in games. These items are then resold for a huge profit while the workers continue to play around the clock to get more items and continue the cycle (Weir, 2004). The VE acts as a renewable resource for marketable items.

E. USER INTERFACE

Companies are beginning to realize that a solid interface with their device means just as much as any other component. With 29.62 million units sold worldwide up to and including the month of June 2008, the Nintendo Wii has outsold both the Xbox (19 million units) and PlayStation 3 (14.4 million units), largely due to a novel interface (Canadian Press, 2008). This propelled the underdog company to become a leader in the console market above corporate powerhouses like Sony and Microsoft. The Apple iPhone is another example of a device using an innovative interface to propel sales in a competitive market. Both the accelerometer and touch screen technologies were

nothing new. Apple managed to incorporate each of these technologies into a device to improve the overall end-user experience.

F. MONITORING THE INTERNET

With all the excitement and innovation occurring in this area, it begs the question – who is watching? Scandals have already rocked the early VEs. Child pornography had a new twist taken on it when users morphed their avatars into child-like forms to allegedly commit sexual acts (Reuters, 2008). A program called Copybot was created that made illegal copies of the virtual objects and textures in Second Life's environment (Reuters, 2006). Virtual speakeasies defied the Second Life gambling ban (Reuters, 2007). With these events unfolding in a VE, the company running the servers where that world exists has been the entity monitoring and policing the environment. This is the same company that is maintaining the hardware, improving the environment, adding features to the application, and keeping the world running. With all the jobs it takes to keep the VE moving, it is likely that monitoring their VE for criminal activity does not fall high on the company's priority list.

Scandals are also nothing new for these types of disruptive use. Internet giants Microsoft and Yahoo were both forced to shut down their user created chat room services because of the amount of child pornography and illegal software being traded (Sullivan, 2005; Reuters, 2003). Even their current configuration is flawed - the Dateline series "To Catch a Predator" showed how men are still soliciting sex from under-age individuals using the

chat room services. Monitoring users of a service is a difficult job. Dateline uses humans to monitor and interact with the suspects (Perverted Justice, 2008). This monitoring by officials is made even tougher by the sheer amount of services available and the subtle ways which they can be used. Applications that are popular one day could be rendered obsolete the next.

G. CLANDESTINE COMMUNICATION

A clandestine operation, as defined in Joint Publication 1-02, is:

> An operation sponsored or conducted by government departments or agencies in such a way as to assure secrecy or concealment. A clandestine operation differs from a covert operation in that the emphasis is placed on concealment of the operation rather than on concealment of the identity of the sponsor. (Joint Chiefs of Staff, 2007)

This definition carries over easily to the Internet, where there are plenty of terrorist organizations utilizing its capabilities to spread their ideas and coordinate distributed efforts.

An innovative method of messaging used by terrorists was a “dead drop” in the drafts feature of an email account. The term dead drop refers to a location used for the clandestine exchange of intelligence information. Email accounts can be used by one terrorist logging into an email account, typing a message and saving it as a draft before logging out. To get the message, all that the recipient terrorist has to do is log into the same account and pull up the draft message. This is beneficial because it does not leave a paper trail and cannot be easily

monitored by authorities. In 2006, it was revealed to the public in court hearings that a British al-Qaeda related terrorist cell was using this email technique to pass messages in preparation for their Madrid bombing, killing 191 people (McLean, 2006). Terrorists are successfully using tools like this to communicate on a daily basis. These tools make them a powerful decentralized force that cannot be taken lightly.

H. MESSAGE PASSING AND WARFARE

Getting information from one individual to another is key in today's world. Everyone from the businessman on Wall Street to the soldier in Afghanistan benefits from a faster communication path. Cell phones, email, websites, and instant messenger are common tools used by the society to convey information. Not surprisingly, terrorists have been using this same set of tools to communicate and coordinate with their fellow members.

Bruce Berkowitz, a research fellow at the Hoover Institution at Stanford University and a senior analyst at RAND, examined the tactics used by both sides during the Global War on Terror. In his book "The New Face of War," he observes that:

> History will not portray Osama bin Laden as a mere terrorist. Rather, instructors at West Point and Annapolis will cite him as one of the first military commanders to use a new kind of combat organization in a successful operation. (Berkowitz, 2003)

This new kind of combat is further described as an interconnected system of autonomous cells, armed with conventional weapons, linked together with a secure

networked communications system for logistics or command and control. Evidence and strong comparisons are drawn between the tactical coordination practices used by warfighters on each side of the fight in countries like Afghanistan do seem to point to a new era of warfare. Osama bin Laden's attacks on September 11 and the military invasion of Afghanistan by U.S. forces both used headquarters located on the opposite side of the planet from the battlefield – GEN Franks from CENTCOM in Tampa, Florida and bin Laden in Tora Bora, Afghanistan. A variety of modes were used to connect the fighters to one another – cellular systems, satellite, fiberoptic, voice, fax, and the Internet. These gave the fighters an encrypted method of accessing a global communications system whose speed and scale had not been seen before (Berkowitz, 2003). Information operations are becoming more and more important to improve the warfighter's situational awareness.

I. EMERGENT GAMEPLAY

Emergent behavior, or emergent gameplay when observed in game environments, describes the development of unforeseen user interactions with a system. These developments can be both beneficial and detrimental to other players in a VE. A beneficial example may be players using a combination of spells in an entirely new, yet effective, way. A detrimental example might be using an avatar's size to block and kill another avatar for their items. "Griefing" is a common term for players whose "sole purpose and intent in any action is to continually upset, aggravate, or otherwise annoy another player" (EA Games,

2008). In many cases, these actions result in warnings or even banning of the account and associated IP address.

This type of emergent use is essential to track. It often determines how a system is used in the mainstream and the degree of popularity that the program will enjoy. If cheating is rampant in a certain application, people will be less likely to play and commit to leveling up the honest way while others are taking shortcuts. An early Massive Multiplayer Online (MMO) game, Ultima Online, experienced massive inflation leading to a currency crisis in 1997. This inflation traced its way back to a flaw allowing users to create duplicate copies of their gold without earning it (Hoglund, 2008).

J. SUMMARY

Virtual Environments have a rich history with many moving parts to consider. All these parts contribute to their sporadic evolution and unpredictable nature. Now that they have been described, the following chapter will discuss methods to pass messages in a clandestine manner.

III. INNOVATION AND EXPERIMENTATION IN VE MESSAGING

A. INTRODUCTION

This chapter will describe the various methods created to pass messages in a clandestine manner. The methods discussed are not an all-inclusive list, but ideas and proof of concepts for innovative techniques. Due to the changing state of VEs and the Internet in general, the techniques described are likely to also change.

B. INNOVATIVE MESSAGING USING VIRTUAL ENVIRONMENTS

VEs present a new method of clandestine communication between two parties. While emails, chat rooms, and instant messages are currently being monitored on one level or another for illegal activity, VEs are largely unmonitored. They depend heavily on user policing of areas and reports of illegal activity to one of the in-world virtual agent authorities. World of Warcraft, a popular VE game, uses high level game avatars to respond to problems, check computer logs, and settle disputes. Second Life uses a similar, in game, reporting system to report violations of their policies. In most examples, the onus rests on the users to report incidents to the game companies and defend their accusations.

Method	Overview	Examples
Visual Signals	Avatar gestures have hidden meanings	Semaphore, Sign Language
State Maintaining Environments	Manipulate an object in the VE that will keep the change after users log out	Adjust virtual rocks, furniture, color patterns
Steganography	Hiding messages in pictures, audio, packets	Place messages in clothing textures, object files
Bots	Automate the logic and control of an avatar	Second Life bots created using AC Tool or C#

Table 1. Clandestine Messaging in Virtual Environments.

The information presented in this section presents new ways to communicate messages from one individual to another in a VE.

1. Visual Signals

One innovative technique unique to VEs lies in the visualization of the avatars. Signals could be passed from one user to another, using sign language-like gestures, semaphore, or tactical hand signals. Even blinking eyelids on an avatar's head could pass a message. Jeremiah Denton, a POW during the Vietnam War proved how effective message passing techniques could be when he spelled out "TORTURE" in Morse code during an interview (Admiral Jeremiah Denton Foundation, 2008).

Second Life makes it easy to set up a simple example of a semaphore technique. Using a free tool called Qavimator, programmers can script a simple animation and save it to a file. All it takes is one individual with the technical skill to create the animations and upload them; after that they can be distributed freely. When uploaded

to the Second Life world, a simple double click will execute this animation. This can even happen while other animations are playing – an avatar can be playing a walking animation while doing a series of semaphore hand gestures with its arms. It doesn't take much imagination to expand this idea to the creation of a comprehensive list of programmable macros that activate animations for each letter and number in the English language.

When these visual signals are used in conjunction with technology, they become even more dangerous. If a pair of gloves that a user wears is able to pick up and mimic the motions they are making, this could easily translate into an animation file or directly into the VE itself. BVH files were originally intended to store motion capture information, their compatibility and use in VEs like Second Life developed later (Gleicher, 1999). A user would not have to take the time to learn a program, just be familiar with the motions of the language.

2. State Maintaining Environments

Virtual worlds often have little to no downtime and maintain state while operating. Using this feature, a party could set up various signals for indicating messages. Signals in virtual sand, movement of virtual rocks to form a pattern, even where avatars park their vehicles or the color of inanimate objects, could each possess some significance for players. While not possessing the data capacity that other systems do, this technique could be used to pass an encryption key or alert other users to a coordinated time. The worldwide access that these systems provide grants anyone with access to the VE the ability to

see these subtle messages. A person in a virtual world could travel the distance between Florida and China in an instant using VE features like in game teleportation or flying.

3. Steganography

Multiple-technique-messaging is made even more dangerous given the tools of virtual environments. For example, a file could be encrypted and hidden inside an image using steganography, stored in a location in the virtual environment, then the decryption password passed via sign language in another area. An innocent picture hanging on the wall in an environment may contain documents, spreadsheets, diagrams, or any combination of these. Disturbingly, instructions for this type of hidden information are present in many books and step by step videos on YouTube.

The only solid defense against this type of message passing is parsing of the image by the VE provider. No method currently exists to crawl a virtual world for pictures or any other type of content as a user. This means that no service exists to search for images. The responsibility lies in the VE server to scan the content being uploaded into the world. A simple script, ten lines written in Ruby, run by the VE server, can perform the necessary operations to scan an image, thereby detecting hidden data using a common steganography technique. It could easily be expanded to detect messages using other scan methods. While the programs to detect such hidden messages may not be complicated, the processing overhead on the VE server could be prohibitive.

The ease with which this type of message passing can be conducted, even on a locked down machine like the Navy Marine Corps Intranet (NMCI), is alarming. Information technology staff that are assigned to scan emails on ships for operational security and pornography do not scan for this type of messaging. If highly technological warfare units defending national security do not monitor for these secret communications, there is little chance that other entities are monitoring for them.

4. Bots

Bots are another powerful tool capable of passing messages. The term bot is derived from the word “robot” and simulates another player in a computer program (Hoglund and McGraw, 2008). These programs have met with mixed reception from the developers of VEs. World of Warcraft has a policy in section 4.B of their terms of service that prohibits the use of “cheats, bots, "mods", and/or hacks, or any other third-party software designed to modify the World of Warcraft experience.” (Blizzard Entertainment, 2007) This expressly prohibits the use of bots to control avatars and gives the company that owns World of Warcraft, Blizzard, the right to terminate the service of any account caught using such programs. There is not the same opposition to programming bots in Second Life that there is in the World of Warcraft. The end user is encouraged to use the Second Life service in any way they choose, as long as it does not violate the terms of use agreement with Linden Labs (Linden Labs, 2008).

Automated control of avatars varies from one VE to another. Two common methods of programming bots are macro editors and APIs. A macro editor is a program that can be used to generate both mouse and keyboard events into a system. Sampling from the pixels on the screen can also be performed to aid in the bot's decision making. An API interfaces directly with the program and allows various functions to be called. These functions control the movement of the avatar, as well as accessing the information going in and out of the avatar's senses.

The benefit of using a bot is that they have the appearance and functions of a normal human using the VE, but without any down time. A bot could be set up to patrol an area indefinitely and deliver a message to a predetermined recipient. Similarly, a patrolling bot could be used to monitor and record communications between avatars. Tie in that monitoring capability with an agent based architecture, and a network of virtual spies comes to life.

C. GAME MANIPULATION

Another factor to be considered is the bendable nature of many computer rules, allowing users to hide messages in remote or restricted areas. The manipulation of these game rules is also useful for monitoring purposes.

Method	Overview	Examples
Camera Controls	Use camera clipping to move into secure and off limit areas	Move the camera through a wall in Second Life and teleport your avatar into a locked room
Game Physics	Alter variables in games to bypass limits and barriers	Speed up avatar movement to make walls useless
Physical Characteristics	Adjust avatar's physical settings in an advantageous way	Change avatar size to one pixel, alter field of view
Building Manipulation	Creative methods of building construction used for spying/message passing	Create one way mirrors, hidden rooms

Table 2. Game Manipulation Methods in Virtual Environments.

1. Camera Controls

In the Second Life VE, the avatar of a user may be denied access to an area using two main methods – locked doors and boundary blocks. Locked doors are virtual doors to rooms that will not open for anyone except a few avatars on the access list. Using a simple camera control provided in the Second Life toolbar, the camera can be moved inside one of these rooms, then an object inside can be clicked on and a "Sit here" command given. The avatar that was denied access at the door will be teleported inside the restricted room. Similarly, boundary blocks are virtual walls extending up into the sky that deny access to any user that does not own the property. The camera can be panned past one of these walls, but the teleportation technique mentioned earlier cannot be used. Virtual land in these areas is actively monitored for the presence of unauthorized visitors. The ease which avatars gain entry

to restricted areas depends on the level of restriction the game itself offers, along with the way that restriction is used.

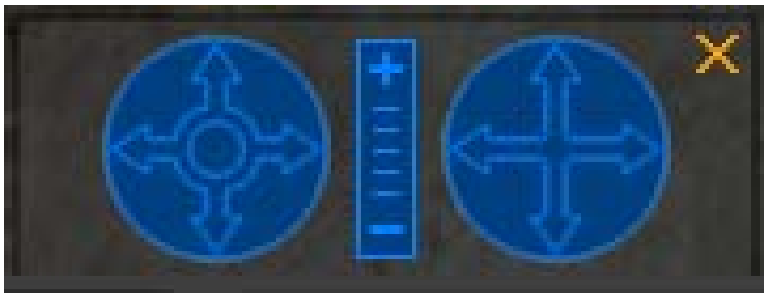

Figure 1. Camera control in Second Life.

Figure 2. Clipping example in Second Life.

2. Game Physics

Delta3D, an open source game and simulation engine, has another example of a loophole present in a virtual system. Avatars in video games are moved using changes to a state variable that describes the avatar location in a virtual world. In the physics engine that Delta3D uses, OpenDynamics, the sample rate for the collision detection

system is slower than the positioning system that changes the state variable. In the words of Delta3D's lead engineer, Erik Johnson:

> It turns out that trying to accurately simulate the physics of a bullet is rather hard to do accurately when dealing with collision detection. Since the size is so small and the speed is so great, you will typically go right through the collided object... (Johnson, 2007)

This means that if an object has a high enough velocity, it will pass through other objects without a collision registering to the system. An avatar could simply fly through a wall if it was traveling fast enough.

No physics engine for a game is perfect – programmers always have to make a sacrifice between processing time and realism. If the sample rate for the collision detection system were increased, it would slow down the VE. This is only one example of one physics engine making these types of sacrifices to improve speed. What other corners could be, or have been, cut that would introduce vulnerabilities into a system?

3. Avatar Physical Characteristics

Users will always attempt to test the bounds of a system – if Second Life did not limit the size of an avatar, there would likely be people flying around the environment that were one pixel large, sneaking through the cracks in doors and gathering information on other's activities. The avatar could be a virtual "fly on the wall," collecting information on others or waiting to pass its own information on to another operative.

Vision is another aspect that can be altered to be advantageous to the user. First person shooter games have been using this trick for years, providing a larger viewing angle and more chance that the user will spot an opponent. It is typically done using a command line interface or by modifying the game code itself (Philips, 2008). Sun's MPK20 provides a selector bar for adjusting the avatar's field of view. Their VE has a default view of 59 degrees that is completely adjustable – even to the point of distortion (Sun Microsystems, 2008). Programmers for a VE will not think of every way a system will be used by a community, that is why they need to keep an eye out for the usage patterns and be ready to fix glitches that users are exploiting.

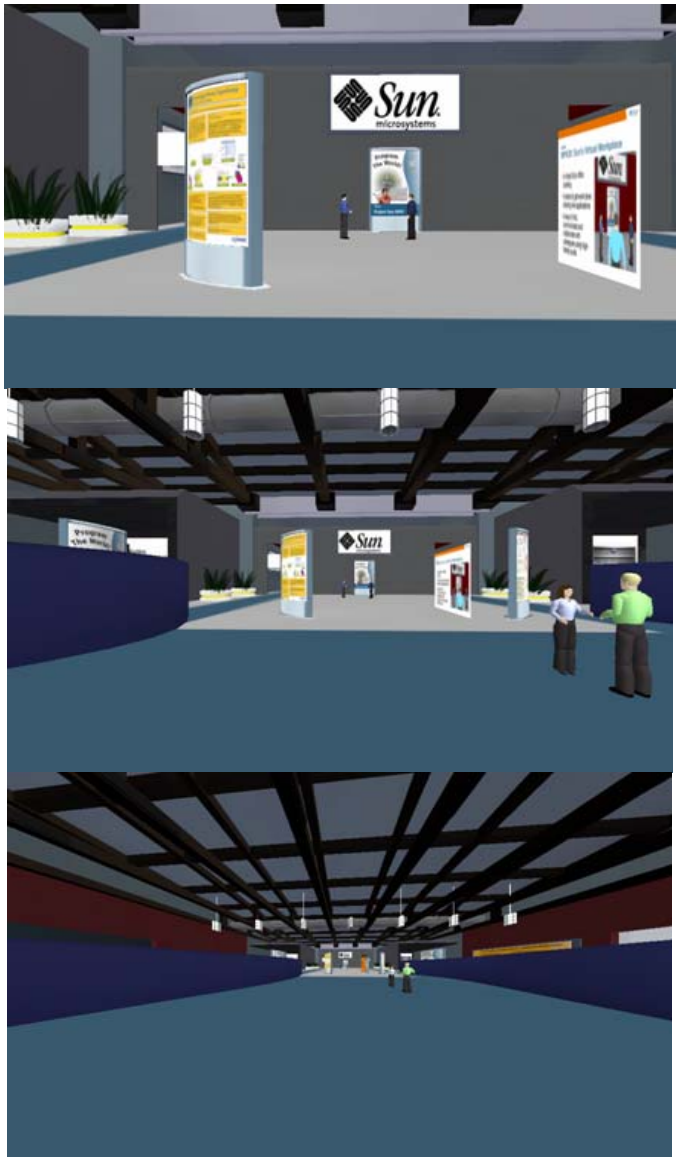

Figure 3. Screen shots of Sun's MPK20 VE with fields of view of 59° (top), 108° (center) and 160° (bottom).

Each of the three screen shots in Figure 3 were taken of the exact same scene and vantage point. Notice how the 59° cuts off the pair of avatars on the right, as shown in the 108° scene. Also notice the level of distortion as the floor, walls, and ceiling take up most of the screen in the 160° view. Players and bots can take advantage of these

increased fields of view, thereby observing more actions taking place in their surroundings.

D. VIRTUAL TRUST

With both civilian and government companies rushing to set up a presence in Second Life, care needs to be taken to prohibit sensitive information in the environment until the extent of that environment is well known. The amount of trust placed in Linden Labs by other companies storing and passing information across their servers is astonishing. The server code has never been released, so Linden Labs has not shown what they do with all the information in their system. Unknown factors include the methods of protection in place for conference calls, encryption level of files passed virtually, and the underlying code for the objects in the room. Linden Labs also states in section 5.3 of their terms of use agreement that “All data on Linden Lab's servers are subject to deletion, alteration or transfer (Linden Labs, 2008).” With companies conducting real world business in VEs and virtual financial dollars linked to real financial dollars, a lot of trust is being placed into these VE systems.

Most companies are not going to take the time to build their own virtual buildings or closely inspect the buildings after they are complete. With that in mind, developers and artists could build rooms with one way mirrors, false walls, hidden passages, and secret rooms. Companies would be having meetings in rooms ripe with surveillance capabilities. This type of surveillance would have an impact on the company’s real world profit by giving away secrets.

This can also be used in conjunction with earlier tools like camera controls to improve clandestine message passing capabilities. If a private room is built with no doors, windows, or entrance of any type, a message can be placed in it by an avatar teleporting in using camera controls. The message can later be read by another avatar by passing through the walls with their own set of camera controls. A normal passer-by would simply consider it a mistake - the owner didn't put an entrance for anyone to go into the room, so it is assumed away as an accident.

E. EXPERIMENTS IN VIRTUAL ENVIRONMENTS

1. Setting up a Virtual Environment

The process of setting up a VE begins with tracking down the source code or installation file for the VE that will be used. The VE used in this experiment was Sun Lab's MPK20 build using the Wonderland VE creator.

Both the source code and executable are available through Sun's website. The source can be compiled with a Java compiler, while the executable contains the files compiled prior to their distribution. Installation of the executable places three programs on the hard drive - Wonderland Server, Wonderland Client, and Wonderland Voice Bridge. The Wonderland Voice Bridge is the first program that needs to be run, followed by the server program. Clients are then able to log into the server and manipulate the environment around them. A client can be based locally on the server machine and remote by using the server's IP address.

2. Passing Information Through Images

The goal of this experiment was to illustrate the ease and accessibility of digital steganography using unclassified sources.

A Google search for “steganography” yields many promising results on the initial page. Perhaps the most interesting are the videos returned. Over forty step-by-step instructional videos exist, all freely available online. They show actual commands of how to hide messages in anything from JPEG format images to WAV audio files. These two to ten minute videos are geared toward the masses that do not have to buy and read a book to learn a new technique. Clicking on one of the videos brings up a computer screen where the actions take place, while a voice adds commentary.

After examining many methods, both in books and online, the simplest way to employ steganography turned out to be in a number of videos on YouTube. This method did not require installation of S-Tools, which is a separate steganography suite. The only requirements are that the computer be running Windows and have a compression utility installed.

The user begins by creating files to be passed. These files can be of any type – spreadsheet, text file, pictures, or any other format of file. Once chosen, these files need to be compressed into a ZIP, RAR, or other archive formatted file. The user must then find a JPEG image to use as a host for the secret files. Once determined, both the image host and compressed archive need to be placed in the same directory. The following command

should be issued in a DOS window without the quotation marks and the period at the end: “copy /B pictureName.jpeg + secretFiles.zip hiddenPicture.jpeg.” This command will instruct the computer to execute a binary file copy of the image host (pictureName.jpeg) and the compressed archive of secret files (secretFiles.zip) to form a completely new, combined picture (hiddenPicture.jpeg). This combined picture file can be opened to show the original image. To gain access to the secret files, all the user has to do is right click on the combined picture file and extract it to the directory. The secret files will decompress from the picture with all of their respective formatting remaining intact. For the user, the difficult part is over – all they need to do is post it online in a VE or send it through a VE document sharing application. The combined image file could even be uploaded as a texture for use in clothing or landscape design.

Innovative twists could be taken on this technique. Passwords could be placed on the zip files or picture files could be hidden inside picture files. Alpha channels in an image are similar to the red, blue, green channels that tell each pixel what color to display. The only difference is that the alpha channel applies to the pixel transparency. Pictures comprised of only a high alpha channel, completely transparent, could be created and used with the digital steganography technique above. Once loaded into a VE, they would be invisible unless someone knew where to look. Appending two files together modifies the size of the original picture. Because the files are picture files, quality could be adjusted to allow the size of the merged file to remain the same. These steganography

techniques will not change the file size of the image if the image is compressed proportionally to the appended file size.

To detect the specific method of steganography discussed above, there is a way to automate scanning by using the Ruby script in Appendix A. The code that defeats it needs to be run on a Windows computer that has a free compression utility installed called 7-Zip and a version 1.86 or higher of the Ruby programming language.

The script will scan through a directory of JPEG files and tap into the command line features of the 7-Zip program to perform extraction attempts on each file. It also uses a regular expression to strip the name of the file from the extension, creating a folder with that name as a title, and will extract all the hidden files found within an image into its corresponding folder.

This method of detection is only one of many possible. It will not catch steganography performed with S-Tools or manually editing bits in the image file. It will, however, catch any files created using the method involving compressed archives and binary copying.

3. Motion Based Messaging

Motion based messaging can be scripted using many tools. Keystrokes could be automated using a macro editor or actual animations programmed into a system. The goal of this experiment was to create an animation that will convey a message with movement.

Second Life was the VE choice for this experiment. Two primary methods of animation are used in the Second Life VE – poseballs and inventory animations. Poseballs are small public floating balls that avatars will teleport to and allow preloaded scripts to animate actions. Inventory animations are private objects retained in an avatar's personal inventory. They contain scripts governing the actions of an avatar and can be played one at a time or in conjunction with other animations. The latter inventory animation is what will be programmed for this experiment.

Qavimator is a free program available for creating animations in Second Life. After installation to a system, the program will launch, and a number of control windows will appear. A three-dimensional window showing a human form will allow the user to select a body part for positioning. Once a body part is selected, its rotation can be adjusted using sliders on the right side of the window. The bottom window contains a series of sliders for each of the selectable body parts. Animation keyframes can be placed at various movement extremities and the computer will interpolate the steps between the movements.

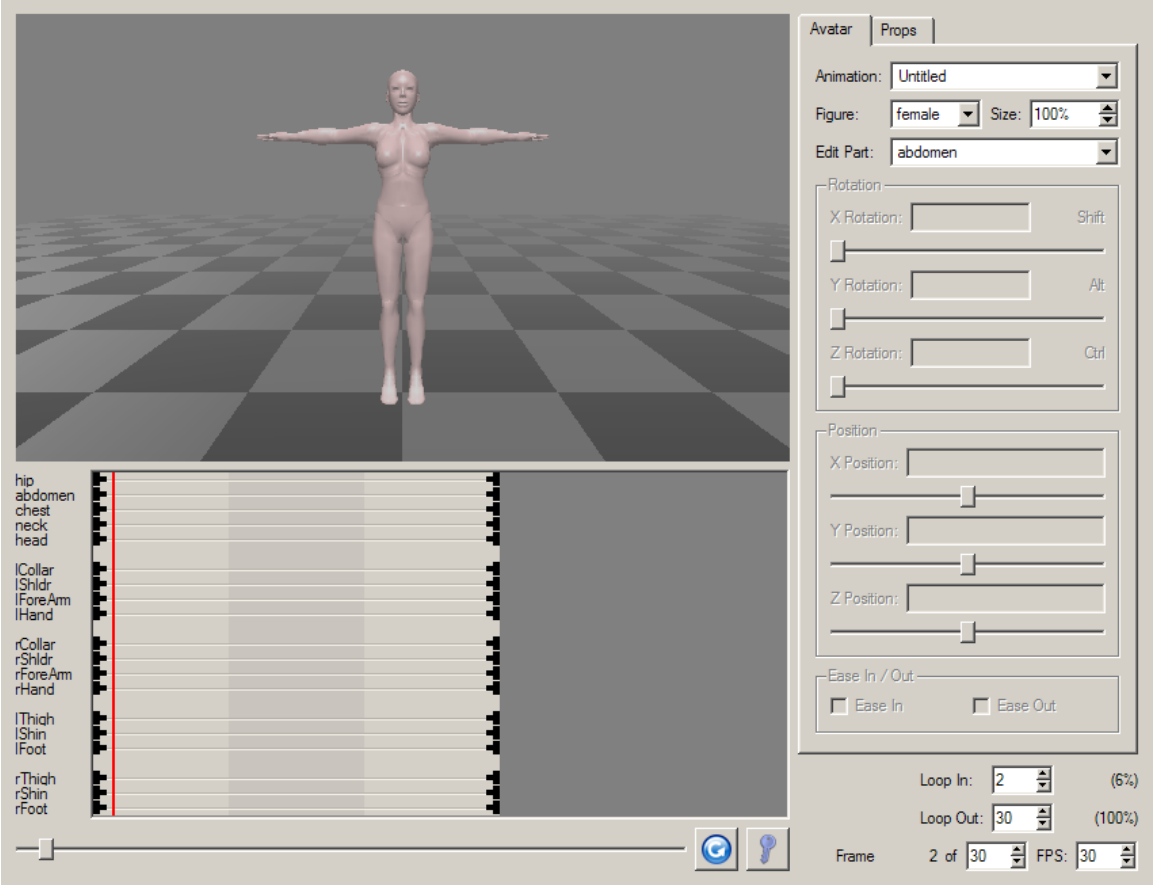

Figure 4. Qavimator screen capture.

The model begins in a standing pose with its arms outstretched and legs together. Semaphore will be the messaging system programmed in this example. If the first letter desired is an R, the semaphore for that particular letter is the arms outstretched with in the position the model is already in. The next desired letter is Y, consisting of the avatar's right arm raised 45 degrees and the left arm remaining stationary. Keyframes for movement will be placed close together and doubled to avoid constant movement by interpolation. A quick transition followed by a longer period of rest in the letter pose is the desired end state for all letters in the word. The letter A will bring the right and left arms down 90 degrees from their

current positions. N, the final letter in the demonstration message, is passed with the left arm moving up 45 degrees and the right arm remaining stationary. The avatar has been successfully programmed to spell the name “RYAN” using semaphore. Second Life requires animations be in the BVH file format to upload, see Appendix B for the source code. Once the animation has been saved in the proper format, the Second Life client should be launched. When the user logs in, there is an option to upload an animation under the file menu. Uploading the animation will place it in the user’s inventory where it can be launched at any time, even while other animations are playing.

This type of message passing is one of the most secure, largely because of a lack of attention. People are not yet in the mindset that nonverbal communication is a possibility using computers. The virtual presence aspect that VEs bring to the table is something not dealt with before.

The data to track this type of messaging would be immense. Even if a server had the storage capacity to record all movements by the avatars inhabiting its servers, the processing required to analyze that amount of data would be enormous. The best place to perform these movements would be out of the way in a secret or off-limits area, so as to not attract attention with all the motions.

4. Programming a Bot

The goal of this experiment was to program a bot that will pass a secret message to a specific avatar and a generic message to others.

A specific programming library exists – libopenmv – that provides the functionality to program an influential bot. Libopenmv, formerly libsecondlife, is a C# API made famous by the Copybot scandal in 2006. It is freely available and can be downloaded using a SVN client. It can then be compiled using a standard installation of Microsoft's Visual Studio. The program to run the bot is created in the same directory and uses function calls to interface with Second Life application, issuing commands. It is also necessary to have a Second Life account that the bot can use to log itself into the system. Source code is available in Appendix C.

F. SUMMARY

This chapter explored methods that could be used to pass clandestine messages in VEs. Visual signals, state maintaining environments, steganography, and bots were all mentioned. Further game manipulation is possible through built-in tools like camera controls that exploit clipping. It can also be accomplished by manipulating values in the virtual environment like the physical characteristics of the avatar (field of view, size, speed) or through holes in the code where the physics engine interacts with the environment itself. Virtual trust was also mentioned, noting that companies doing business in VEs need to examine the office space for virtual bugs, false walls, and hidden areas.

Now that these methods have been discussed, ideas to help in monitoring and tracking of such activities will be explored.

IV. ANALYSIS AND MONITORING OF VIRTUAL ENVIRONMENTS

A. INTRODUCTION

By discussing examples of message passing techniques in Chapter III, analysis can take place on those methods in an attempt to develop detection and tracking systems. Monitoring on the Internet is tricky due to a multitude of things, including issues of legality and the mechanics of actual message detection. This chapter will shed light on possible solutions to improve detection and discuss other issues that pertain to VE monitoring systems.

B. HONEYWORLDS

The Honeyworld concept was developed after the techniques mentioned in Chapter III for message passing were researched and attempted. It gets its name from the similar Honeypot network idea – decoy servers or systems set up to gather information regarding an attacker or intruder into a network.

A Honeyworld is a server based VE, much like Second Life, where the government would retain full ownership of the source code. The code for the server application would be closed source while the client application would be open source. Retaining rights to the open source code would allow the government to know exactly what was behind the programs that are accessing and transmitting their sensitive data. It would also allow the government the freedom to change and modify whatever portions they desired.

The government networks are already segmented into information levels – unclassified, secret, top secret. This segmentation allows trend analysis to take place on the unclassified network while conducting business on the secret and top secret networks. New attacks can be observed, illegal activities can be monitored, and new technologies can be tested on the unclassified network. The secret network would be where the reliable applications and technologies become commonplace. Three major benefits exist with a Honeyworld system: collaboration, monitoring, and training.

1. Collaboration

Improved collaboration is a selling point for many of the VE systems in the current market. Sun illustrated this feature with their MPK20 VE – their OpenOffice.org word processor could work as both a standalone application or as an imbedded application to be manipulated in the VE. This kind of compartmentalization and interoperability maximizes the cohesion of the communication systems while still retaining flexibility. A style of allowing the users to choose what functionality they received is ideal for forward deployed units that have lower bandwidth than shore installations. Even among deployed units, ships for example, there are different bandwidth capabilities. Cruisers have the bandwidth to support VTC conferences and are beginning to use that feature with a third party set of VTC applications. Frigates, on the other hand, have little more bandwidth than a dial-up connection. If the applications were compartmentalized, the cruisers could be running a VTC application in the VE while the frigates ran

another part of the VE software like chat or voice over IP. Interoperability would allow parts of the applications to work together – the audio for the VTC could be all the frigate receives, but it would be enough to allow it participation in the meeting.

2. Monitoring

Monitoring refers to setting up observation servers to host VE worlds on unclassified networks. When users log in, their actions are capable of being observed for various studies. Usage trends can be mapped, showing things like avatar placement and popular new applications. The files being transferred between users pass through the server, so they can be monitored for content. New ways of using the VE can also be observed through monitoring and analysis. Areas for authorities to emphasize on can be tested for effectiveness. Traditional Honeypots have shown value in this area – new viruses are commonly caught and observed on the Honeypot systems in networks. World of Warcraft (WoW), a game based VE, uses a client installed program to monitor users of its service. This program actually sends the names and identification information of all the other open windows on a user's computer back to the creators of WoW (Hoglund and McGraw, 2008). A similar user created tool is called Thottbot. Thottbot is an online database of WoW information, available to users that also install the Thottbot plugin. The plugin acts like a virtual assistant giving statistics and timing for attacks and weapon drops while also feeding back information to the servers about the scenarios the user is encountering. In this way, it acts as a distributed monitoring tool. As new threats are

encountered, information about specific threats are sent back to the central database for analysis, classification, and storage.

3. Training

Training is another major benefit of such a system. With a standardized application, training and simulation suites can be incorporated to improve simulations. Shipboard training is typically done with live teams onboard ships reading scripts to sailors. If a VE is used, chat and VTC clients can be used like they were in real-life operations. Locations can be virtually mapped and visualized for ports around the world. Down the road, full simulations could be run from the Combat Information Center to train watch teams during downtime. Junior officer training has already taken steps in this direction. CDs are distributed to all new officers onboard ships with PowerPoint training files for instruction. While at their surface warfare school, officers spend a considerable amount of time piloting virtual ships using head mounted displays in the VE simulators. This shipboard simulation package would help push training out to the people who need it the most – the deployed forces.

C. HONEYWORLD LEGAL ISSUES

This type of research is filled with legal questions that are largely unanswered. Questions of monitoring, entrapment, and reasonable expectations of privacy all come to mind. Honeypots have faced very similar, if not identical, debate over their use in a public environment.

Not all aspects of the Honeyworld idea have these issues and, when used in the proper setting, much may be learned from them. Academic institutions could run studies after using them and the military could analyze Honeyworld use within their own installation intranets.

D. PEER TO PEER THREAT

Peer to peer communication bypasses the client/server monitoring capabilities that the Honeyworld is designed to monitor. Building in a solid distributed monitoring capability would be the best way to combat this type of a threat. Users would periodically report information about themselves and others in the VE.

E. CRITICISM OF VIRTUAL ENVIRONMENTS

Why would people want to use a VE over an instant messenger client or chat room/web page combination? The same can be asked about cell phones. People could have phones that use a quick and efficient command line interface. The reason for the success of devices like the Blackberry is not that it invented email, web access, or voice phone calls. It brings those communication tools together in one easy to use interface.

Similar combinations are occurring in the desktop computing field. Applications like instant messenger had different proprietary screen names that couldn’t be used across services. AOL, Yahoo, IRC, and MSN all had their own protocols that were not compatible.

Newer software bridged the gap between these devices. Downloadable applications like Trillian and Pidgin (formerly Gaim) allowed cross service communication. Meebo is another web based browser application that helps with this cross platform communication.

Virtual Environments will be the tool that bridges the current gaps between applications. Military bases may one day possess VEs on their intranets that allow people to have a quick VTC with a base legal officer for a power of attorney while both parties are editing the virtual document in real time. People wouldn't have to leave their desk if they needed to get information from their personal file in base administration. They could enter a virtual queue that would alert them when they were next up to speak with a representative and examine their record making gestures to point out information. This type of system works perfectly with the Navy's minimal manning initiative on ships like the Littoral Combat Ship (LCS). With most of the sailors pulled to provide support from shore facilities, effective services for distance collaboration need to be developed.

F. SUMMARY

This chapter described the basic layout of a potential method of monitoring VEs – the Honeyworld concept. A Honeyworld could track emergent behavior, act as a better tool for collaboration among military units, and support immersive training applications.

V. SUMMARY AND CONCLUSIONS

A. OVERVIEW

This thesis focused on a threat analysis of the methods of passing messages in VEs.

By exploring traditional methods of message passing using computers and then adapting new ones for VEs, potential tools for clandestine message passing were identified. New techniques that VEs bring to the table were discussed and experimented with. Interception of these new techniques was examined. Finally, new ideas about monitoring were applied to the VE settings.

B. ARCHITECTURE FOR A METAVERSE

The information presented in this thesis is only a beginning. VEs will continue to change and grow, developing new features while constantly increasing their potential use for clandestine message passing.

It is difficult to predict exactly how VEs will be used in the future. By examining the current use of information like web pages and email servers, some light may be shed on what the future holds. Small business may turn to VEs run by larger companies. Large business may end up running their own VEs instead of placing that kind of trust with other companies. The idea of a language allowing various VEs to communicate and avatars to cross over has already been conceived. If it takes off, an entire metaverse could be created comprised of smaller VEs.

Somewhat like a “roaming identity,” users could jump from VE to VE using their avatars as simply as they surf the Internet.

Personal users of computer services are accustomed to trusting that all their information is safe. Users of Gmail, Hotmail, and Yahoo mail do not own the servers where their emails are stored. Personal information like passwords, credit card numbers, and bank accounts are all in many people’s email accounts, all without a second thought from them on who has their information. When compatibility with these databases increases, VEs will have access to massive repositories of user data. Pair that amount of data with tighter integration into social networking sites, and the picture of massive, all knowing VEs begins to take shape.

C. MOBILE DEVICES

Mobile devices like cell phones, portable gaming systems, and even MP3 players add a new dimension to message passing. The hardware involved in the field of mobile devices is rapidly evolving. Two devices that are capable of using both 802.11 wireless networking and ad-hoc networking are the PSP and DS. The CPU processing speed for the PSP is 333MHz while the DS has two CPUs with speeds of 67MHz and 33MHz. Processor speeds for the Apple iPhone are rumored to be somewhere around the 600MHz range. With the less evolved hardware capable of ad-hoc networking, it will not be long until common cell phones are able to set up their own local networks. When these smaller networks

are created, they will not use the conventional communications grid. New monitoring capabilities will need to be created to counter this emerging threat.

D. FOLLOW-ON RESEARCH TOPICS

The following is a list of topics for future research:

- Exploration of legal issues and their impact on VEs. Whose jurisdiction does it fall under? What level of monitoring can law enforcement do?
- Modify a VE to allow easy extraction of data for analysis. What information is easiest to track?
- Analyze the most important factors that agencies need to monitor in a VE. Is there a way to track and store this data? How do you analyze the amount of data involved in a VE for signs of clandestine messaging?
- The potential exists for a mobile/ad-hoc system to be implemented using a number of devices on the market. As mentioned, the PSP and Nintendo DS both have wireless networking capabilities. These capabilities do not require a base station to establish communication between units. Nodes, the handheld units, can directly connect to peer nodes. This differs from the cell phone/wireless computer schemes, and can be much more powerful if a message needs to be sent secretly among local entities.
- Economic ties to VEs. With virtual economies tied to real economies, what are the effects? Should Second Life players be allowed to sell their virtual land but World of Warcraft players not allowed to sell items? How can rules be better adapted to manage these scenarios?
- Create a working Honeyworld.
- Conduct a compatibility analysis of using a VE with current military communications systems. Can the hardware support a VE with current equipment? If not, when will the military be able to? Is there a better way to fit the pieces together?

- Conduct an in-depth investigation based on how agencies are currently monitoring terrorists, child predators, and drug trafficking. Are there any unknown or obscure tools that could improve VE monitoring? Could an agent based system of avatars improve monitoring?

E. A BRIGHT FUTURE

Virtual Environments are still in a relatively early state of development – they have yet to become the common tool for communication or conducting business. Mobile devices are beginning to see simple VEs that can be played over the cellular networks. Innovative interfaces like the Nintendo Wii and Apple iPhone are the first devices in a series that will reinvent the way humans interface with devices. When this is paired with the growing power of computing, commonplace video hardware, and increasing bandwidth it shows enormous potential for technologies like VEs. If the government can harness this collaboration potential, it could prove extremely beneficial as a system.

Government agencies have often been reactionary with their network security and monitoring. This attitude gives free reign to individuals during the infancy of many products, exploiting systems in innovative ways to avoid detection. If the government takes a larger role in many of the VE developments, it will be ahead of the game when the newest tools arrive to the public.

The threats are real and could be disastrous if ignored. Not all message passing will be caught, but that doesn’t mean it should be ignored entirely. A strong

government presence in these VEs will act as a deterrent and allow identification of potential problems as they emerge.

F. THE CONTINUING PROCESS

Much like conventional warfare, cyber warfare is a continuing process. For each advancement made by either side, the tactics, techniques and procedures will update accordingly. The capability bar will be set higher and new clandestine messaging techniques will emerge. An excellent model to help explain this and get people's minds in the correct frame helped to revolutionize maneuver warfare for the Marines – the OODA Loop. Colonel John Boyd created the OODA (Observation, Orientation, Decision, Action) Loop as a method of enabling adaptation. By observing actions taking place in these VEs, trends can be established. Orienting the necessary thinking and efforts towards the problems observed is the next step. After establishing the proper orientation, a decision is necessary to select the most important courses of action to engage the most important threats. The action phase implements this decision and then the process repeats with the result feeding into the observation step (Clark, 2004).

The key to maintaining dominance in this area is having well-rounded individuals supporting the subject matter experts. The experiments in this thesis required knowledge of various programming languages, computer graphics, three-dimensional modeling, human computer interaction, and artificial intelligence. Being able to

take information and look at it through the lens of a bigger picture will help the government one step ahead of the terrorists in this constantly changing environment.

APPENDIX A. RUBY STEGANOGRAPHY DETECTION SOURCE CODE

```
#Author: Ryan Rippeon
fileHandle = []
done = false
Dir["*.jpg"].each do |f|
  strippedextension = f.gsub(/\./,'_')
  commandLine = '"C:\Program Files\7-Zip\7z.exe" e '+f+' -
o.\\'+strippedextension
  puts "Executing: "+commandLine
  puts `#{commandLine}`
  puts "done"
end
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. BVH ANIMATION CODE

```
HIERARCHY
ROOT hip
{
     OFFSET 0.000000 0.000000 0.000000
     CHANNELS 6 Xposition Yposition Zposition Xrotation
Zrotation Yrotation
     JOINT abdomen
     {
          OFFSET 0.000000 3.422050 0.000000
          CHANNELS 3 Xrotation Zrotation Yrotation
          JOINT chest
          {
               OFFSET 0.000000 8.486693 -0.684411
               CHANNELS 3 Xrotation Zrotation Yrotation
               JOINT neck
               {
                    OFFSET 0.000000 10.266162 -0.273764
                    CHANNELS 3 Xrotation Zrotation
Yrotation
                    JOINT head
                    {
                         OFFSET 0.000000 3.148285 0.000000
                         CHANNELS 3 Xrotation Zrotation
Yrotation
                         End Site
                         {
                              OFFSET 0.000000 3.148289
0.000000
                         }
                    }
               }
               JOINT lCollar
               {
                    OFFSET 3.422053 6.707223 -0.821293
                    CHANNELS 3 Yrotation Zrotation
Xrotation
                    JOINT lShldr
                    {
                         OFFSET 3.285171 0.000000 0.000000
                         CHANNELS 3 Zrotation Yrotation
Xrotation
                         JOINT lForeArm
                         {
```

```
                              OFFSET 10.129278 0.000000
0.000000
                              CHANNELS 3 Yrotation
Zrotation Xrotation
                              JOINT lHand
                              {
                                   OFFSET 8.486692 0.000000
0.000000
                                   CHANNELS 3 Zrotation
Yrotation Xrotation
                                   End Site
                                   {
                                        OFFSET 4.106464
0.000000 0.000000
                                   }
                              }
                         }
                    }
               }
               JOINT rCollar
               {
                    OFFSET -3.558935 6.707223 -0.821293
                    CHANNELS 3 Yrotation Zrotation
Xrotation
                    JOINT rShldr
                    {
                         OFFSET -3.148289 0.000000 0.000000
                         CHANNELS 3 Zrotation Yrotation
Xrotation
                         JOINT rForeArm
                         {
                              OFFSET -10.266159 0.000000
0.000000
                              CHANNELS 3 Yrotation
Zrotation Xrotation
                              JOINT rHand
                              {
                                   OFFSET -8.349810
0.000000 0.000000
                                   CHANNELS 3 Zrotation
Yrotation Xrotation
                                   End Site
                                   {
                                        OFFSET -4.106464
0.000000 0.000000
                                   }
```

```
                              }
                         }
                    }
               }
          }
          JOINT lThigh
          {
               OFFSET 5.338403 -1.642589 1.368821
               CHANNELS 3 Xrotation Zrotation Yrotation
               JOINT lShin
               {
                    OFFSET -2.053232 -20.121670 0.000000
                    CHANNELS 3 Xrotation Zrotation Yrotation
                    JOINT lFoot
                    {
                         OFFSET 0.000000 -19.300380 -1.231939
                         CHANNELS 3 Xrotation Yrotation
Zrotation
                         End Site
                         {
                              OFFSET 0.000000 -2.463878 4.653993
                         }
                    }
               }
          }
          JOINT rThigh
          {
               OFFSET -5.338403 -1.642589 1.368821
               CHANNELS 3 Xrotation Zrotation Yrotation
               JOINT rShin
               {
                    OFFSET 2.053232 -20.121670 0.000000
                    CHANNELS 3 Xrotation Zrotation Yrotation
                    JOINT rFoot
                    {
                         OFFSET 0.000000 -19.300380 -1.231939
                         CHANNELS 3 Xrotation Yrotation
Zrotation
                         End Site
                         {
                              OFFSET 0.000000 -2.463878 4.653993
                         }
                    }
               }
          }
```

```
}
MOTION
Frames:   120
Frame Time:    0.033333
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
0.000000 43.528519 0.000000 0.000000 0.000000 0.000000
```

APPENDIX C. C# BOT SOURCE CODE

```
//Portions from libsecondlife website
//Author: Ryan Rippeon
using System;
using System.Collections.Generic;
using System.Text;
using libsecondlife;

namespace MyFirstBot
{
   class MyFirstBot
   {
       public static SecondLife client = new SecondLife();

       private static string first_name = "######"; //First
       private static string last_name = "######"; //Last
       private static string password = "######";
//Password
       private static string secretMessage = "We will
attack at
dawn."; //Secret Message to Pass
       private static bool sentMessage = false;

       public static void Main()
       {
           client.Network.OnConnected+= new
NetworkManager.ConnectedCallback(Network_OnConnected);
           if (client.Network.Login(first_name, last_name,
password,
"My First Bot", "Your name"))
           {
               Console.WriteLine("I logged into Second
Life!");
           }
           else
           {
               Console.WriteLine("I couldn't log in, here
is why: " +
client.Network.LoginMessage);
           }
           // put this somewhere when you want to start
processing
instant messages
           client.Self.OnInstantMessage += new
```

```
AgentManager.InstantMessageCallback(Self_OnInstantMessage);
       }


       //(...) then define the Self_OnInstantMessage method

       static void Self_OnInstantMessage(InstantMessage im,
Simulator sim)
       {

             //there are a variety of InstantMessageDialog
choices..
MessageFromObject and MessageFromAgent would be the two
most common
             if (im.Dialog ==
InstantMessageDialog.MessageFromAgent)
             {
                 client.Self.InstantMessage(im.FromAgentID,
im.Message, im.IMSessionID);
                 Console.WriteLine("User: " +
im.FromAgentName + "
With ID: " + im.FromAgentID + " just sent me an IM.");
                 //send them an instant message back (this
thing will
copy any message the bot recieves in an IM)
                 if (im.FromAgentID ==
"########-####-####-####-############" && sentMessage ==
false)
                 {

client.Self.InstantMessage(im.FromAgentID,
im.FromAgentName+" you are my secret contact. Here is my
message: ",
im.IMSessionID);

client.Self.InstantMessage(im.FromAgentID,
secretMessage, im.IMSessionID);
                     sentMessage = true;
                 }
             }
             for (int i = 0; i < 10000; i++) ;
             if (im.Message == "exit")
                 client.Network.Logout();
         }
```

```
        static void Network_OnConnected(object sender)
        {
            Console.WriteLine("I'm connected to the
simulator, going
to greet everyone around me");
            //client.Self.Chat("Hello World!", 0,
ChatType.Normal);
            Console.WriteLine("Now I am going to log out of
SL...Goodbye!");
        }
    }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Admiral Jeremiah Denton Foundation. “Biography of Jeremiah A. Denton,Jr.. (2008). ”http://www.dentonfoundation.org/jeremiahdenton.html (accessed July 31, 2008).

America’s Army. “Game Intel.” (January 2008). http://www.americasarmy.com/intel/features.php (accessed August 27, 2008).

Berkowitz, Bruce. The new face of war: How war will be fought in the 21st century. Glencoe: Free Press, 2003.

Berman, Art. “Will Web 3.0 be in 3D?” (April 2007). http://displaydaily.com/2007/04/10/will-web-30-be-in-3d/ (accessed April 4, 2008).

Blizzard Entertainment. “World of Warcraft: Terms of Use Agreement.” (January 2007). http://www.worldofwarcraft.com/legal/termsofuse.html (accessed July 31, 2008).

Boyer, Brandon. “PSP firmware update unlocks full CPU.”(June 2007). http://www.gamasutra.com/php-bin/news_index.php?story=14439 (accessed July 31, 2008).

Canadian Press. “Wii nears one million consoles in Canada, drives Nintendo profits.” (July 2008). http://canadianpress.google.com/article/ALeqM5g6yL5Scx xBnEFr3ddagwwglmLSig, (accessed July 31, 2008).

Clark, Donald. “OODA: Observe, Orient, Decide, & Act.” (October 2004). http://www.nwlink.com/~Donclark/leadership/ooda.html (accessed August 27, 2008).

Cole, Eric. “Hiding in plain sight: Steganography and the art of covert communication.” Indiannapolis: Wiley Publishing, 2003.

Coll, Steve and Susan Glasser. “Terrorist Turn to the Web as Base of Operations.” (August 2005). http://www.washingtonpost.com/wp-dyn/content/article/2005/08/05/AR2005080501138_pf.html (accessed July 31, 2008).

Das, Tapas K, Gurminder Singh, Alex Mittchell, P.Senthil Kumar, and Kevin McGee. (1997). “Developing social virtual worlds using NetEffect.” IEEE Enabling Technologies: Infrastructure for Collaborative Enterprises, 148-154.

Delta3D. “About Delta3D.” (July 2008). http://www.delta3d.org/article.php?story=20041105154425816&topic=about (accessed July 31, 2008).

Disney. “Disney Pirates of the Caribbean Online.” (January 1, 2008). http://apps.pirates.go.com/pirates/v3/#/game_info/about.html (accessed August 21, 2008).

EA Games. “Customer Support: What do you consider "grief tactics," anyway?” (January 2008). http://support.ea.com/cgi-bin/ea.cfg/php/enduser/std_adp.php?p_faqid=1418&p_created=994178867&p_sid=p9IXL3si&p_lva=&p_sp=cF9zcmNoPTEmcF9zb3J0X2J5PSZwX2dyaWRzb3J0PSZwX3Jvd19jbnQ9NjAmcF9wcm9kcz03LDc2OSZwX2NhdHM9NTY1LDcwNSZwX3B2PTIuNzY5JnBfY3Y9Mi43MDUmcF9wYWdlPTM*&p_li=&p_topview=1 (accessed July 31, 2008).

Johnson, Erik. “Questions and Answers.” (May 2007). http://www.delta3d.org/forum/viewtopic.php?forum=14&showtopic=9029 (accessed July 31, 2008).

Gamecubilce. “Nintendo DS Fact Sheet.” (August 2004). http://www.gamecubicle.com/hardware-nintendo_ds_spec_sheet.htm (accessed July 31, 2008).

Gleicher, Michael. “Biovision BVH.” (March 1999). http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html (accessed July 31, 2008).

Hearne, John. “NPSNET: Physically Based, Autonomous, Naval Surface Agents.” Naval Postgraduate School Thesis, 1993.

Herodotus. The Histories. Trans. Robin Waterfield. Oxford: Oxford UP, 1998.

Hoglund, Greg and McGraw, Gary. (2008). Exploiting Online Games: Cheating Massively Distributed Systems. New York: Addison-Wesley.

Joint Chiefs of Staff. (January 2006). “Information Operations.” Joint Publication 3-13. Washington, DC.

Joint Chiefs of Staff. (July 2007). “Department of Defense Dictionary of Military and Associated Terms.” Joint Publication 1-02. Washington, DC.

Know Your Enemy: The Honeynet Project. Boston: Addison-Wesley, 2004.

Linden Labs. “Currency Exchange.” (January 1, 2008). http://secondlife.com/whatis/currency.php (accessed July 31, 2008).

Linden Labs. “Mainland Pricing & Fees.” January 1, 2008. http://secondlife.com/land/pricing.php (accessed July 31, 2008).

Linden Labs. “Second Life Terms of Use Agreement.” (July 31, 2008). http://secondlife.com/corporate/tos.php (accessed July 31, 2008).

McLean, Renwick. “Madrid suspects tied to e-mail ruse.” (April 28, 2006). http://www.iht.com/articles/2006/04/27/news/spain.php (accessed July 31, 2008).

Microsoft. “Get Windows Vista: System Requirements.” (July 31, 2008). http://www.microsoft.com/windows/windows-vista/get/system-requirements.aspx (accessed July 31, 2008).

Miller, John and Page, Scott. Complex Adaptive Systems: An Introduction to Computational Models of Social Life. Princeton: Princeton UP, (2007).

Miller, Nick. “Banking meltdown infects the virtual world.” (January 15, 2008). http://business.theage.com.au/business/banking-meltdown-infects-the-virtual-world-20080114-1ly3.html (accessed April 4, 2008).

Moss, William. “Report From Singapore.” (August 24, 1997). http://www.imagethief.com/files/documents/singnote/singnot7.html (accessed August 27, 2008).

MTV. “vMTV – Registration - Pick a Show!” (January 1, 2008). http://www.vmtv.com/pick_show.html (accessed August 27, 2008).

Parent, Rick. Computer Animation: Algorithms and Techniques. New York: Morgan Kaufmann, (2008).

Perverted Justice. “Frequently Asked Questions.” (January 1, 2008). http://www.perverted-justice.com/index.php?pg=faq (accessed April 4, 2008).

Philips, John. “TF2, Episode 2, Portal, & Steam Client Updates Released.” (June 9, 2008). http://planethalflife.gamespy.com/fullstory.php?id=151566 (accessed July 31, 2008).

Reuters. “Microsoft to Shut Down Chat Rooms.” (September 23, 2003). http://www.wired.com/culture/lifestyle/news/2003/09/60567 (accessed April 4, 2008).

Reuters, Adam. “Outcry as ‘Copybot’ threatens copyright protection.” (November 14, 2006). http://secondlife.reuters.com/stories/2006/11/14/outcry-as-copybot-threatens-copyright-protection/ (accessed April 4, 2008).

Reuters, Eric. “Ageplay sim eyes new grid.” (March 11, 2008). http://secondlife.reuters.com/stories/2008/03/11/ageplay-sim-eyes-new-grid/ (accessed April 4, 2008).

Reuters, Eric. “One small step for avatars, one giant leap for avatar-kind?” (June 13, 2008). http://secondlife.reuters.com/stories/2008/06/13/one-small-step-for-avatars-one-giant-leap-for-avatar-kind/ (accessed July 31, 2008).

Reuters, Eric. “Virtual speakeasies defy Second Life gambling ban.” (August 14, 2007). http://secondlife.reuters.com/stories/2007/08/14/virtual-speakeasies-defy-second-life-gambling-ban/ (accessed July 31, 2008).

Schach, Stephen. Object-Oriented and Classical Software Engineering. Boston: McGraw-Hill, 2002.

Sullivan, Bob. “Yahoo chat choice signals Internet shift.” (June 23, 2005). http://www.msnbc.msn.com/id/8336384/ (accessed July 31, 2008).

Sun Microsystems. “Project Wonderland User FAQ.” (July 31, 2008). http://wiki.java.net/bin/view/Javadesktop/ProjectWonderlandEndUserFAQ#Can_I_fly_in_Wonderland (accessed July 31, 2008).

Terdiman, Daniel. “A brief history of the virtual world.” (November 9, 2006). http://news.cnet.com/A-brief-history-of-the-virtual-world/2008-1043_3-6134110.html (accessed August 27, 2008).

Tode, Chantal. “Sears, IBM launch Second Life virtual store.” (January 8, 2007). http://www.dmnews.com/Sears-IBM-launch-Second-Life-virtual-store/article/94053/ (accessed April 4, 2008).

Weir, Laila. “Boring Game? Outsorce It.” (August 24, 2004). http://www.wired.com/entertainment/music/news/2004/08/64638 (accessed April 4, 2008).

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Gurminder Singh
   Naval Postgraduate School
   Monterey, California

4. CDR Joseph Sullivan, USN
   Naval Postgraduate School
   Monterey, California

5. Tom Frey
   Wyle Laboratories
   Virginia Beach, Virginia

6. Charles Villamarin
   Sandia National Laboratories
   Albuquerque, New Mexico

7. ADM James Hogg, USN (Ret)
   CNO Strategic Studies Group
   Newport, Rhode Island

8. Glen Donovan
   Central Intelligence Agency
   Washington, DC

9. Kevin Farrell
   NIOC, Suitland
   Suitland, Maryland

10. Robert Sandoval
    USSTRATCOM, JIOWC/J3
    San Antonio, Texas

11. Ms. Rosemary Wenchel
    DISL OSD OUSD
    Washington, DC

12. LCDR James Caroland
NIOC Suitland, N5 Special Liaison/Tech Ops
Suitland, Maryland

13. Steven Iatrou
Naval Postgraduate School
Monterey, California

14. Chairman, Dept. of Information Sciences
Naval Postgraduate School
Monterey, California

15. Program Officer, JC4I Curriculum
Naval Postgraduate School
Monterey, California

16. Raymond Buettner
Naval Postgraduate School
Monterey, California